ממ"ן 15 – שלד פתרון

הקורס : 20471 (ארגון המחשב) 2023א

שאלה 1 (30%)

1. **הסבר מילולי**

היות ונרצה לעצור את התכנית, במידת הצורך, בשלב הכתיבה לזיכרון, נרצה ששני האוגרים VAL ו ADDR יהיו ממוקמים בשלב הMEM.
הפקודות שעלולות להשתנות בעת הוספת הרכיב הם פקודות מסוג store בהן אות הMemWrite דלוק. בשאר הפקודות לא תתבצע כתיבה לזיכרון ולכן לא ניתקל בבעיה.
מלבד שני האוגרים, נצטרך להיעזר ב:

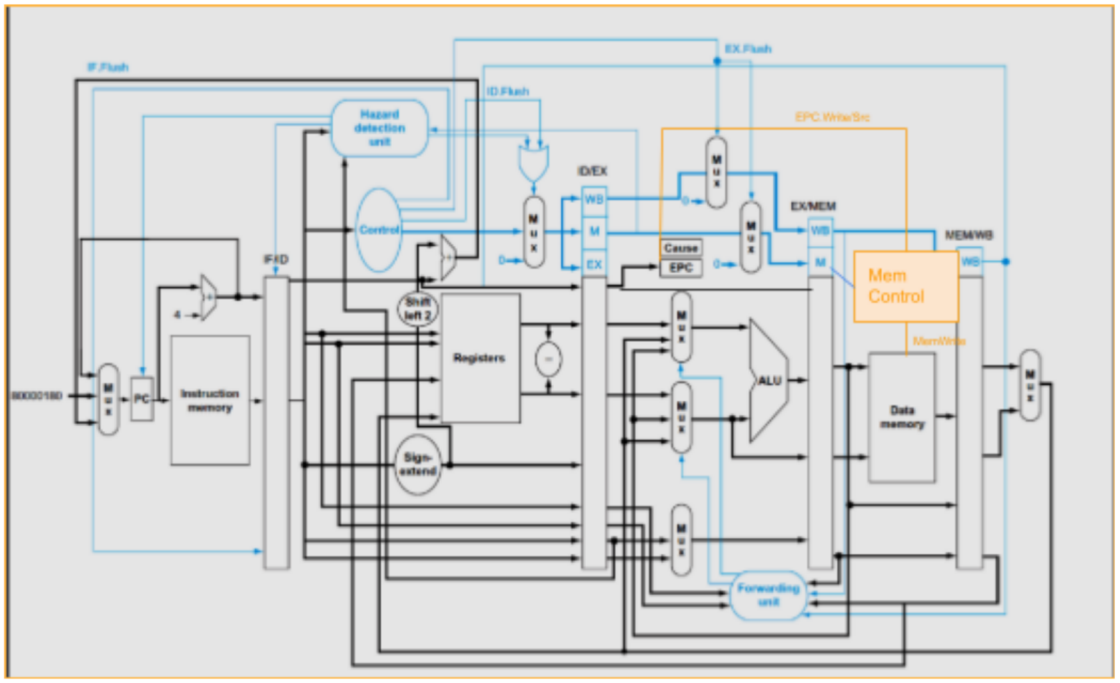
- רכיב השוואה comparator המשווים בין הערך הנמצא באוגר val לבין הערך העומד להיכתב בזיכרון (EX/MEM.Data)
- רכיב השוואה comperator המשווה בין הערך באוגר addr לבין הכתובת אליה נרצה לכתוב (EX/MEM.ALUOut)
- שער AND בעל שלוש כניסות - שוויון בערך הval, שוויון בערך הaddr, וסיגנל MemWrite דלוק.
- כתובת PC+4 אותה יש לקחת איתנו משלב הEXE, כלומר להוסיף 32 סיביות נוספות לאוגר EX/MEM.

כאשר שלוש התנאים הללו מתקיימים, נרצה לכתוב לאוגרים EPC וCause (הערך שייכתב לCause יהיה קוד שגיאה שרירותי, נניח 15).
גם אם תתקיים שגיאה בפקודה הנמצאת בשלב הEXE, אין הדבר משנה, שכן הפקודה הכותבת לזיכרון מופיעה לפניה ברצף הפקודות.
לסיכום, השינויים הנדרשים בחומרה הם:

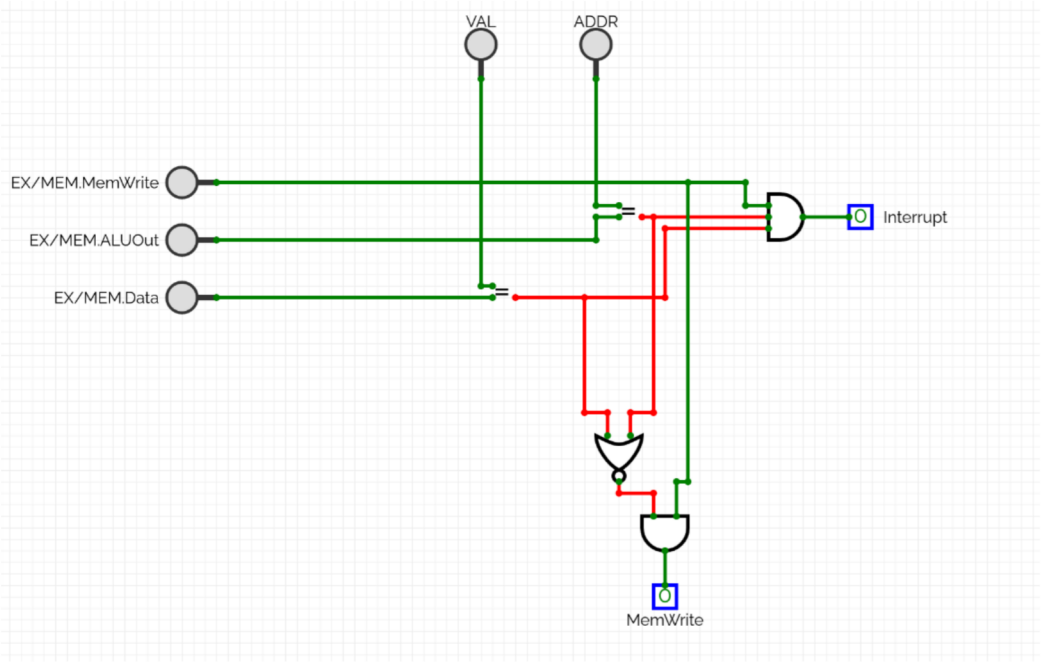
- קביעת הלוגיקה להתקיימות השגיאה
- מניעת כתיבה לזיכרון בפועל - חיווי כניסת MemWrite
- כתיבת הכתובת הרלוונטית לאוגרים EPC וCAUSE

איננו נדרשים לאפס את אותות הWB (ביצוע Flush) משום שהם ממילא 0 בפקודות store

הצגת תרשים 4.66 עם התוספות הנדרשות בנתיב הנתונים



כאשר רכיב הMem Control ייראה כך:



1. **מה הסיבה להכנסת אפסים בכל אחת משתי הכניסות לשער ה OR ? האם יתכן ששתי הכניסות יקבלו 1 בו זמנית? במידה וכן תתארו תרחיש כזה, במידה ולא הסבירו מדוע.**

לשער הOR שתי כניסות - אחת הבאה מהHDU והשנייה הבאה מה Control - אות ID.Flush.
הסיבה לכך שמכניסים אפסים בכניסה מהHDU היא משום שהרכיב לא זיהה סיכון נתונים הדורש הכנסה בועה (סיכון load-use).
הסיבה לכך שמכניסים אפסים בכניסה הבאה מה Control היא עקב כך שרכיב הבקרה לא זיהה חריגה שמקורה בסוג פקודה שגוי (או כל מקור אחר שניתן לזהות בשלב הID).
יכול להתקיים מצב בו שתי הכניסות יקבלו 1 בו זמנית. נתבונן בקטע ההוראות הבא:

$$lw\ \$8,\ 0(\$6)$$

$$xxx\ \$8,\ \$6, 27$$

ונניח שהפקודה השנייה קודדה מעזרת האסמבלר לשורת קוד בעלת opcode לא תקין (שהרי אחרת לא תתקבל חריגה כלל). מצד אחד, רכיב הcontrol יזהה חריגה שמקורה בפקודה לא מזוהה. מצד שני, רכיב הHDU יזהה תלות של אוגר המקור rs (בכל סוגי הפקודות) בערך הנקרא מהזיכרון. לכן שתי הכניסות יקבלו 1 בו זמנית.

2. **תארו תרחיש בו אות IF.Flush יקבל את הערך 1.**

אות IF.Flush יקבל את הערך 1, למשל, כאשר ביצענו חיזוי שגוי של branch. הפקודה שקראנו לא אמורה להתרחש ולכן יתקבל אות IF.Flush.

3. **תארו תרחיש בו אות ID.Flush יקבל את הערך 1.**

אות ID.Flush יקבל את הערך 1, למשל, כאשר נקראה מהמעבד הוראה שאינה מוגדרת.
אז רכיב הבקרה ישחרר אות ID.Flush.

4. **תארו תרחיש בו אות EX.Flush יקבל את הערך 1.**

אות EX.Flush יקבל את הערך 1, למשל, כאשר נקלט arithmetic overflow. הפקודה המבוצעת גרמה לשגיאת Overflow ולכן יתבצע Flush.

5. **האם יכול להתקיים תרחיש בו שלושת האותות בסעיפים ג-ה יקבלו 1 בו זמנית? במידה וכן תתארו תרחיש כזה, במידה ולא הסבירו מדוע.**

יכול להתקיים תרחיש המתואר לעיל. למשל, כאשר מתבצעת שגיאת overflow, נרצה שלא לבצע את הפקודה היוצרת את השגיאה וגם את הפקודות **שאחריה**, שכן אז נקבל סיכון נתונים חמור! לכן, כל הפקודות שנמצאות בצנרת לא יבוצעו, ונפנה ישר לטיפול בבעיה.

שאלה 2 (25%)

1. **מהו סוג זיכרון זה ?**

מדובר בזיכרון מסוג $2-way\ set\ associative$, באופן דומה לתרשים 5.18. ההבדל הוא שסוג הזיכרון המתואר במטלה הוא בעל 2 בלוקי מידע בכל שורה (סט) ולא 4, ובהתאם משתמשים ב-2 רכיבי השוואה ובמרבב $2 \rightarrow 1$ ולא ב-4 רכיבים ובמרבב המתאים.

2. **מה כמות הנתונים שניתן לאפסן בזיכרון זה ביחידות של סיביות ? הראו חישובכם.**

כמות הנתונים היא 1024 סיביות

חישוב:

מספר בלוקי-המידע שניתן לאפסן בזיכרון זה הוא 2 בלוקים בכל סט, עבור 16 סטים, ובסך הכל 32 בלוקים.

בכל בלוק שמורה מילה אחת (נתון לנו ע"פ חלוקת הביטים בכתובת), ובסך הכל 32 מילים.

בכל מילת-מידע יש 4 בתים (שוב ע"פ חלוקות הביטים), ובסך הכל 2^7 בתים.

בכל בית יש 8 סיביות, ובסך הכל 2^{10} סיביות, כלומר 1024.

3. **מה המספר הכולל של סיביות בזיכרון זה (כולל valid bit ו tag dirty bit)? הראו חישובכם**

המספר הכולל הוא: 1920 סיביות

חישוב:

כפי שפירטנו בסעיף הקודם, בזיכרון שלנו שמורים 32 בלוקי מידע.

ע"פ חלוקת הביטים בכתובת בכל בלוק אנחנו שומרים 26 ביטים לתווית (תג), בנוסף לשני הביטים valid bit, dirty bit.

מכאן שכמות הסיביות הנלווים למידע השמור בזיכרון תהא נתונה ע"י $32 \times 28 = 896$.

סיביות נלוות אלה נוספות לסיביות המידע שאת כמותן חישבנו בסעיף הקודם, ונקבל $1024 + 896 = 1920$.

4. **מה תפקיד ה dirty bit ? והאם הוא יעיל לנתוני הזיכרון שבתרשים ? נמקו תשובתכם.**

תפקיד ה dirty bit הוא לשמור מידע על האם בוצעה כתיבה לבלוק מאז שנשמר ב cache, על מנת לא לבצע כתיבות מיותרות לרכיב זיכרון נמוך יותר בהיררכיה כאשר משתמשים בwrite-back ולא כותבים ישירות לזיכרון הראשי כמו ב write-through.

ככל שגודל הבלוק (במילים) גדול יותר, כך גדל הסיכוי שנכתוב לזיכרון מילות-מידע שלא שונו בפועל, דבר המבזבז זמן ומשאבים ואינו יעיל.

לעומת זאת, נתוני הזיכרון שבתרשים שמורים בבלוקים המכילים מילת-מידע אחת, ולכן בהכרח לא יתבצעו כתיבות מיותרות לזיכרון והשיטה יעילה מאוד לנתוני הזיכרון שבתרשים.

5. **במידה והייתה פניה לכתובת 0x2578aaaa בבתים בזיכרון הראשי לאיזו שורה(סט) בזיכרון המטמון תתבצע גישה על מנת לחפש האם נתון נמצא במטמון ?**

מספר השורה: 10.

חישוב:

נרשום בבינארי את הכתובת הנתונה: 0010 0101 0111 1000 1010 1010 1010 1010

או, אם נרווח בהתאם לחלוקת הסיביות בזיכרון, 10 1010 0010010101111100010101010 10.
ערך סיביות האינדקס יהיה $1010bin$, או 10 בבסיס עשרוני, ולכן תתבצע גישה לסט מספר 10 במטמון.

שאלה 3 (45%)

1. **חלוקת הסיביות (index tag וכו) בזיכרון זה. אין צורך להציג byte offset, שכן נתונות לנו הכתובות <u>במילים</u>.**

בזיכרון הנתון לנו בסעיף א (מיפוי ישיר או one-way set associative), ישנם שמונה בלוקים (נדרשות $\log_2 8 = 3$ סיביות אינדקס) ובכל בלוק שמונה מילים (נדרשות 3 סיביות block-offset). לכן, החלוקה תהא:

29-3-3-2=21	3	3
Tag	Index	Block offset

סמנו החטאה או פגיעה לכל כתובת : (השורה הראשונה מולאה לדוגמא)

(Addr (word	Hit or miss	index	Tag	(DATA(word	הערות
12	m-valid	1	0	Mem[8-15]	
33	m-valid	4	0	Mem[32-39]	
67	m-valid	0 (8 mod 8)	1	Mem[64-71]	
15	**hit**	1	0	Mem[8-15]	
73	m-tag	1 (9 mod 8)	1	Mem[72-79]	בזיכרון היו מילים 8-15
56	m-valid	7	0	Mem[56-63]	
89	m-valid	3 (11 mod 8)	1	Mem[88-95]	
13	m-tag	1	0	Mem[8-15]	בזיכרון היו מילים 72-79
68	**hit**	0 (8 mod 8)	1	Mem[64-71]	
44	m-valid	5	0	Mem[40-47]	
46	**hit**	5	0	Mem[40-47]	
60	**hit**	7	0	Mem[56-63]	
63	**hit**	7	0	Mem[56-63]	
45	**hit**	5	0	Mem[40-47]	
54	m-valid	6	0	Mem[48-55]	
8	**hit**	1	0	Mem[8-15]	
79	m-tag	1 (9 mod 8)	1	Mem[72-79]	בזיכרון היו מילים 8-15

התוכן הסופי של זיכרון המטמון:

index	Valid	Tag	Data (words)
0	1	1	Mem[64-71]
1	1	1	Mem[8-15]
2	0	?	?
3	1	1	Mem[88-95]
4	1	0	Mem[32-39]
5	1	0	Mem[40-47]
6	1	0	Mem[48-55]
7	1	0	Mem[56-63]

שיעור הפגיעות הוא 7/17 או 41.17647059%.

2. **חלוקת הסיביות בזיכרון זה :**

כל בלוק מכיל 8 מילים, כלומר נדרשות 3 סיביות offset
זיכרון fully associative - אין סיביות אינדקס
נותרו 26 סיביות tag

26	0	3
Tag	Index	Block offset

צורת סימון LRU צד שמאל זה way 0 צד ימין way 3.
משמעות המספר 0 זה ה Least Recently Used) LRU)
אילו משמעות המספר 3 זה ה Most Recently Used) MRU)
נניח איתחול LRU שרירותי מ 0 ל 3 כלומר (0,1,2,3)
אפיון כל כתובת בסדרת הגישות:

Address (word)	Hit or miss	Tag	Data(word)	Way/Block	LRU (start 0,1,2,3)	הערות
12	m-valid	1	Mem[8-15]	0	3,0,1,2	
33	m-valid	4	Mem[32-39]	1	2,3,0,1	
67	m-valid	8	Mem[64-71]	2	1,2,3,0	
15	**hit**	1	Mem[8-15]	0	3,1,2,0	
73	m-valid	9	Mem[72-79]	3	2,0,1,3	
56	m-tag	7	Mem[56-63]	1	1,3,0,2	פונו מילים 32-39
89	m-tag	11	Mem[88-95]	2	0,2,3,1	פונו מילים 64-71
13	**hit**	1	Mem[8-15]	0	3,1,2,0	
68	m-tag	8	Mem[64-71]	3	2,0,1,3	פונו מילים 72-79
44	m-tag	5	Mem[40-47]	1	1,3,0,2	פונו מילים 56-63
46	**hit**	5	Mem[40-47]	1	1,3,0,2	
60	m-tag	7	Mem[56-63]	2	0,2,3,1	פונו מילים 88-95
63	**hit**	7	Mem[56-63]	2	0,2,3,1	
45	**hit**	5	Mem[40-47]	1	0,3,2,1	
54	m-tag	6	Mem[48-55]	0	3,2,1,0	פונו מילים 8-15
8	m-tag	1	Mem[8-15]	3	2,1,0,3	פונו מילים 64-71
79	m-tag	9	Mem[72-79]	2	1,0,3,2	פונו מילים 56-63

התוכן הסופי של זיכרון המטמון:

Way/Block	Valid	Data (words)	Tag	LRU
0	1	M[48-55]	6	1
1	1	M[40-47]	5	0
2	1	M[72-79]	9	3
3	1	M[8-15]	1	2

שיעור הפגיעות בזיכרון זה : 5/17 או 29.411764705%

3. חלוקת הסיביות בזיכרון זה :

בכל בלוק יהיו 2 מילים, כלומר נדרשת ספרת offset יחידה.
ישנם שמונה סטים, כלומר סך הכל נידרש ל3 ספרות אינדקס.

29-3-1-2=23	3	1
Tag	Index	Block offset

אפיון כל כתובת בסדרת הגישות: (נניח אכלוס ראשוני לבלוק אפס LRU=0)

Addr (word)	Hit or miss	index	Tag	Way/Block	Data (words)	הערות
12	m-valid	6	0	0	Mem[12-13]	
33	m-valid	0	2	0	Mem[32-33]	
67	m-valid	1	4	0	Mem[66-67]	
15	m-valid	7	0	0	Mem[14-15]	
73	m-valid	4	4	0	Mem[72-73]	
56	m-valid	4	3	1	Mem[56-57]	
89	m-tag	4	5	0	Mem[88-89]	יוחלפו מילים 72-73
13	**hit**	6	0	0	Mem[12-13]	
68	m-valid	2	4	0	Mem[68-69]	
44	m-valid	6	2	1	Mem[44-45]	
46	m-valid	7	2	1	Mem[46-47]	
60	m-tag	6	3	0	Mem[60-61]	יוחלפו מילים 12-13
63	m-tag	7	3	0	Mem[62-63]	יוחלפו מילים 14-15
45	**hit**	6	2	1	Mem[44-45]	
54	m-valid	3	3	0	Mem[54-55]	
8	m-tag	4	0	0	Mem[8-9]	יוחלפו מילים 88-89
79	m-tag	7	4	1	Mem[78-79]	יוחלפו מילים 46-47

התוכן הסופי של זיכרון המטמון:

		Block 0			Block 1		
Set	**LRU**	**Valid**	**Tag**	**Data (words)**	**Valid**	**Tag**	**Data (words)**
0	0	1	2	Mem[32-33]	0	?	?
1	0	1	4	Mem[66-67]	0	?	?
2	0	1	4	Mem[68-69]	0	?	?
3	0	1	3	Mem[54-55]	0	?	?
4	0	1	0	Mem[8-9]	1	3	Mem[56-57]
5	0	0	?	?	0	?	?
6	1	1	3	Mem[60-61]	1	2	Mem[44-45]
7	1	1	3	Mem[62-63]	1	4	Mem[78-79]

שיעור הפגיעה במקרה הזה הוא : 2/17 או 11.7647058824%